

REMARKS

Claims 1-44 are pending. Claims 34 and 41 have been canceled. Independent claims 1, 10, 16, 25, 31, 38, 43 and 44 were rejected under 35 U.S.C. 103(a) as being unpatentable over the iPlanet Application Server 6.0 White Paper from Sun Microsystems (hereinafter the Sun reference) in view of Reynolds. It should be noted that Sun is also the assignee of the present application. Although the claims are believed patentable in their current form, the independent claims along with dependent claim 4 and 19 have been amended to facilitate prosecution.

The independent claims have been amended to recite "a first object-oriented component residing at a first server invoking a second object-oriented component residing at a second server." The Examiner argues that the Sun reference teaches a method for a first component to invoke a second component in an object-oriented computing environment. The material cited by the Examiner, however, only describes asynchronous messaging associated with Java Messaging Services (Sun: Figure on page 23). Asynchronous messaging is not invocation of one object-oriented component by another.

The Sun reference however, does describe asynchronous invocation of an object-oriented component by another object-oriented component. The Sun reference describes the Corba Executive Server. "The Corba Executive Server (CXS) or Bridge process allows for independent Java-enabled clients (Rich Clients) to communicate directly to components based on the Enterprise JavaBeans architecture and hosts on a Java Server." (Sun: page 22, last line)

The Corba Executive Server is described in the background of the present application. More specifically, "Common Object Request Broker (CORBA) provides a framework for asynchronous messaging. CORBA is described in "Common Object Request Broker: Architecture and Specification: CORBA 2.4.1", November 2000, the entirety of which is hereby incorporated by reference for all purposes. CORBA requires derived types, however, that introduce unnecessary complexity into the programming model" (page 2, lines 29-34).

The techniques of the present invention recognize some of the drawbacks of CORBA. "CORBA provides a set of interfaces that is independent of where the object is located or what language the object is implemented in. However, support for asynchronous object invocations in

CORBA is a complex patchwork solution placed on top of the synchronous object invocation framework.” (page 2, line 29 – page 3, line 5)

“To allow asynchrony, CORBA redefines the interface with derived types as follows:

```
interface A {  
    void foo(float, foo_callback c);  
}  
    interface foo_callback {  
        void handlereturnforfoo(int);  
        void handleexception1(exception1);  
        void handleexception2(exception2);  
        void handleexception3(exception3);  
    }
```

 (page 3, lines 12-23).

One method must handle the return for foo and separate methods are written for each exception. It should be noted that the number of methods in the interface grows with the number of application specific exceptions. The use of derived types in CORBA unnecessarily increases complexity of the programming model.” (page 3, lines 26-30)

The techniques and mechanisms of the present invention provide an elegant and efficient technique for performing asynchronous invocation of an object-oriented component by other object-oriented components.

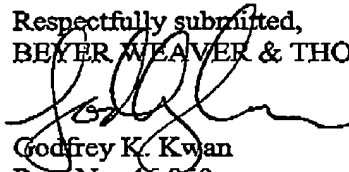
The Object Management Group (OMG) was founded in 1989 and created CORBA in 1991. CORBA is a relatively successful standard that has been widely adopted. However, despite the long felt and unmet need for more efficient asynchronous object-oriented component invocation, this need has not been believed met until the conception of the techniques of the present invention.

The independent claims variably recite registering an exception listener on an asynchronous proxy associated with the second component, wherein the exception listener is stateless and is operable to handle a plurality of types of exceptions from a plurality of different components. Neither CORBA nor any other reference teach or suggest an object invocation

model where an exception listener is registered on an asynchronous proxy. It would not be appropriate to combine CORBA and any other reference to teach such a recitation because CORBA does not support such an exception listener. Other exception listeners noted by the Examiner are not "exception listener object-oriented components" invoked by other components, as also recited in the independent claims.

In light of the above remarks relating to independent claims, the remaining dependent claims are believed allowable for at least the reasons noted above. Applicants believe that all pending claims are allowable and respectfully request a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER, WEAVER & THOMAS, LLP



Godfrey K. Kwan
Reg. No. 46,850

P.O. Box 70250
Oakland, CA 94612
(510) 663-1100